ЗАПАДНО-СИБИРСКИЙ НАУЧНЫЙ ЦЕНТР

Сборник Международной
научно-практической конференции

АКТУАЛЬНЫЕ И ПЕРСПЕКТИВНЫЕ НАПРАВЛЕНИЯ РАЗВИТИЯ НАУЧНО-ТЕХНОЛОГИЧЕСКОГО ПРОГРЕССА

30 января 2020 г.
Кемерово

ЗАПАДНО-СИБИРСКИЙ НАУЧНЫЙ ЦЕНТР

АКТУАЛЬНЫЕ И ПЕРСПЕКТИВНЫЕ НАПРАВЛЕНИЯ РАЗВИТИЯ НАУЧНО-ТЕХНОЛОГИЧЕСКОГО ПРОГРЕССА

Сборник материалов
Международной научно-практической конференции

30 января 2020 г.

г. Кемерово

УДК 44.01 + 331 + 61 + 338 + 622 + 009 + 50 + 004 + 62 + 7 + 8 + 691 + 551.521 + 63 + 656 + 34
DOI 10.5281/zenodo.1319336
ГРНТИ 12.09.11
ББК 1

Актуальные и перспективные направления развития научно-технологического прогресса: сборник материалов Международной научно-практической конференции (30 января 2020 г.), – Кемерово: ЗапСибНЦ, 2020 – 102 с.

ISBN 978-5-9909594-6-0

Сборник материалов конференции содержит научные статьи отечественных и зарубежных авторов, посвященные актуальным и перспективным направлениям развития научно-технологического прогресса.

Предназначен для ученых, преподавателей, аспирантов и студентов высших и средних специальных учебных заведений, научно-технических работников и специалистов в области технических, естественных и гуманитарных наук, информационных технологий, горного дела, геодезии, строительства и архитектуры, сельского хозяйства, пищевой промышленности, экономики, юриспруденции.

Ответственность за аутентичность и точность цитат, названий и иных сведений, а также за соблюдение законов об интеллектуальной собственности несут авторы публикуемых статей.

Мнение оргкомитета и редколлегии может отличаться от мнения авторов статей, опубликованных в сборнике научных трудов.

Материалы публикуются в авторской редакции.

ОГЛАВЛЕНИЕ

ГЕОДЕЗИЯ, СТРОИТЕЛЬСТВО И АРХИТЕКТУРА

ГУМАНИТАРНЫЕ НАУКИ

ЕСТЕСТВЕННЫЕ НАУКИ

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

ПСИХОЛОГИЯ И ПЕДАГОГИКА

СЕЛЬСКОЕ И ЛЕСНОЕ ХОЗЯЙСТВО

навыки, как если бы они находились в реальной жизни.

3. Маркетинг в образовании. У технологий AR есть огромный потенциал для маркетинга и рекламы, даже в сфере образования. Ряд университетов в США уже используют AR-туры, чтобы увеличить количество учащихся и помочь новым студентам пробираться по кампусу.

Например, Общественный колледж округа Бивер, штат Пенсильвания, использует дополненную реальность для предоставления различных типов контента, включая видео, аудио и цифровые публикации. Таким образом, приложение предоставляет интересный и информативный способ исследовать кампус. Приложение также имеет элементы геймификации, чтобы сделать его еще более увлекательным.

Как мы видим, дополненная реальность в образовании обладает огромным потенциалом, который еще предстоит раскрыть. С текущим внедрением мобильных технологий и недавними достижениями в области аппаратных средств AR становится все более доступным и широко используемым. Поэтому сейчас самое время сделать первые шаги в этом направлении.

Список литературы:

1. Петрова О. Дополненная реальность для целей образования. Электронный ресурс. URL: https://goo.gl/N2rhaF

2. Maria Aleksandrova Augmented Reality in Education. Электронный ресурс. URL: https://dzone.com/articles/augmented-reality-in-education

FOR DATA SECURITY USING SYMMETRIC ENCRYPTION ALGORITHM

Temirbekova Zh.E. – senior lecture,
Kazakh national university named after al-Farabi,
Kazakhstan, Almaty city

Abstract

This article describes some of the programming and debugging techniques developed using the Mbed platform running on the BLE Nano microcontroller [1]. C ++ was used as the primary language instead of C to improve programming efficiency. This turned out to be a good choice, but it led to an increase in the size of the program. Thus, much of the optimization was spent on reducing the size of the program and data. The main purpose of this article is to describe the Advanced Encryption Standard (AES) symmetric encryption algorithm, its security and complexity, using the ARM Mbed special encryption library [2].

Keywords

Internet of Things (IOT), BLE Nano kit microcontroller, Mbed platform, AES cryptography, Homomorphic encryption.

In the recent decade, microcontroller is getting popular application in portable devices, embedded system and mobile platform due to its integrated architecture, rich functionalities and increasing processing power.

The Bluetooth Low Energy (BLE) microcontroller used is RedBearLab's BLE Nano. According to their website, the BLE Nano kit is the smallest Bluetooth 4.1 Low Energy (BLE) development board in the market [3]. The core is Nordic nRF51822 (an ARM Cortex-M0 SoC plus BLE capability) running at 16MHz with ultra-low power consumption.

Developing a Bluetooth Smart enabled 'appcessory' (accessory device + companion application) is easier than ever. You can quickly produce prototypes and demos target for IoT

and other interesting projects. BLE Nano could operate under 1.8V to 3.3V; therefore it works with a lot of electronic components.

The BLE Nano kit was chosen not only for its low power consumption but also for its ease of use. The board comes with the MK20 USB board and several software development options, including an online option that was used for this work. To upload code, the BLE Nano is attached to the MK20 USB board and plugged in to a USB port where upon a drag-and-drop interface facilitates uploading code to the board over the USB connection. The BLE Nano can then be removed from the MK20 USB board and tested in a breadboard circuit. Many existing libraries and examples as well as available Android and IOS applications made this microcontroller a good choice to start with.

To prepare the BLE Nano, header pins were soldered onto the Nano and MK20 USB board. Mbed was the online compiler and IDE used in this work, so code from the mbed RBL Nano was used to run a blink test on the BLE Nano and boatload the MK20.

Programming with mbed

Mbed is an online compiler and API (Application Program Interface) that was used to code this work. BLE Nano kit microcontroller is supported by the Mbed's hardware platform. Applications for the Mbed platform can be developed using the Mbed online IDE, a free online code editor and compiler. Only a web browser needs to be installed on the local PC, since a project is compiled on the cloud, i.e. on a remote server, using the ARMCC C/C++ compiler [4-5].

Mbed OS provides the Mbed C/C++ software platform and tools for creating microcontroller firmware that runs on IoT devices. It consists of the core libraries that provide the microcontroller peripheral drivers, networking, build tools and test and debug scripts.

The many available platform libraries are easy to use and well designed, especially for starters, resembling (vaguely) the simplicity you may have enjoyed when using Arduino libraries. Also, Mbed encourages and eases sharing with others your own libraries, or any piece/snippet of code. You have to register a username in the platform from the very beginning that will allow you to store your projects online, comment, share code and participate in the community forums comfortably [7]. Compiling and programming your projects is really easy. You can literally have the led blink test application running in a few minutes.

Mbed has a dedicated team which develops and maintains a very nice Bluetooth Low Energy API. You can rest assured that using this API for learning to develop BLE *appcessories* will be much easier (and faster) than trying to do so by using directly the BLE libraries (Soft Device) provided by Nordic, because they are already integrated into the Mbed platform The BLE API includes also several application examples.

On the not-so-positive side of Mbed platform, you have zero control over the compiler/linker. This has proven to be a showstopper for our own BLE development. At the time of this writing, within Mbed you are only allowed to develop BLE applications for the nRF51822 along with the S110 Soft Device, which only supports peripheral mode. Hence, you can quickly start to develop applications with Mbed which only require the Peripheral Role. But if you want to develop an application requiring the Central Role, you will have to switch to a full-fledged GCC (or Keil or IAR) C/C++ compiler.

Coding is done in C++, and the compiler provides a programming environment in which to write the code as well as the function of compiling the code into an executable file for the platform.

AES algorithm overview:

AES is based on a design principle known as a substitution-permutation network, a combination of both substitution and permutation, and is fast in both software and hardware. AES is a variant of Rijndael which has a fixed block size of 128 bits, and a key size of 128,

192, or 256 bits. By contrast, the Rijndael specification *per se* is specified with block and key sizes that may be any multiple of 32 bits, with a minimum of 128 and a maximum of 256 bits.

AES Algorithm consists of 2 Main Parts:

1- Encryption or Decryption Process:

In each round, the AES uses the four following operations:

•SubBytes: Each byte of the array is transformed using a nonlinear substitution box called the AES S-Box. The S-Box in the AES has been carefully constructed and the cipher uses only one S-Box throughout the encryption.

•ShiftRows: Is a transposition step which ensures that the last three rows of the array are shifted by a different number of byte positions.

•MixColumns: Mixes each column in the array to create even more diffusion.

•Addkey: Using bitwise XOR, each byte of the array is mixed with a byte of a sub-key material, also called round-key. The sub-key is made by "key expansion" and is derived from the main cipher key using a Rijndael key-schedule.

Decryption process: it is reverse of encryption process in every step which means the decryption first circle is the tenth circle of the encryption and it uses the invers functions of MixColumns, SubBytes, ShiftRows and us you can assume the keys arrangement and reversed too as it starts with Addkey10 instead of Addkey0 as it was in the encryption process.

2-Key generation.

It involves RotWord, SubBytes and XOR bitwise operation to generate enough keys for each circle In the Encryption, Decryption process.

AES-CBC buffer encryption and decryption. Length should be a multiple of the block size (16 bytes). Upon exit, the content of the IV (initialization vector) is updated so that you can call the function same function again on the following block(s) of data and get the same result as if it was encrypted in one call. This allows a "streaming" usage. If on the other hand you need to retain the contents of the IV, you should either save it manually or use the cipher module instead. parameter ctx - the AES context to use for encryption or decryption, parameter mode - MBEDTLS_AES_ENCRYPT or MBEDTLS_AES_DECRYPT, parameter length - length of the input data, parameter iv -initialization vector (updated after use), parameter input-buffer holding the input data, parameter output-buffer holding the output data, return 0 if successful, or MBEDTLS_ERR_AES_INVALID_INPUT_LENGTH.

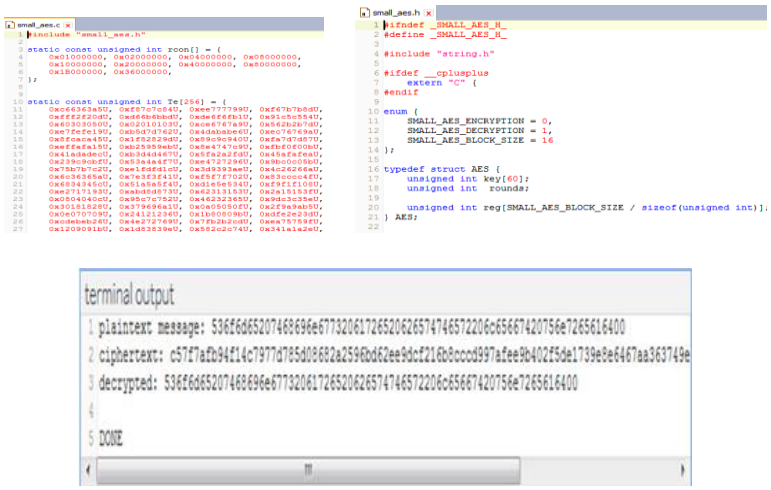

Figure 1 - Performance of AES ciphers in Mbed platform.

AES also has one advantage over other encryption algorithms. Advantage of AES - Cracking a 128 bit AES key with a state-of-the-art supercomputer would take longer than the presumed age of the universe. And Boxcryptor even uses 256 bit keys. As of today, no practicable attack against AES exists. Therefore, AES remains the preferred encryption standard for governments, banks and high security systems around the world.

Security is fundamental for the successful rollout of the Internet of Things. Edge nodes are currently the weakest link in ensuring IoT security and the protection of cryptographic key. The best way to achieve lockdown is by protected hardware. It is the only way to keep those keys and other secrets away from prying eyes. An IoT device can only be as secured as its weakest link.

In this paper the implementation AES and fully homomorphic encryption in a binary number ring algorithm. Has given it more encryption power thus makes it harder for anyone to hack the ciphered information and decrypted it.

References:

4. BLE Nano. http://redbearlab.com/blenano/.

5. Jose Angel, BLE Nano hardware development kit for Bluetooth Low Energy, - 2015

6. S. Aguilar, R. Vidal, C. Gomez, Opportunistic Sensor Data Collection with Bluetooth Low Energy. *Sensors* - 2017, - P. 159.

7. C. Gomez, J. Oller, J. Paradells, Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology. Sensors - 2012, - P. 11734–11753.

8. J. Nieminen, T. Savolainen, M. Isomaki, B. Patil, Z. Shelby, C. Gomez, IPv6 over BLUETOOTH(R) Low Energy; RFC 7668; IETF: Fremont, CA, USA, - 2015